

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-78938

(43) 公開日 平成10年(1998) 3月24日

(51) Int.Cl. ⁹	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16 15/177	3 7 0		G 0 6 F 15/16 3 7 0 N 4 0 0 S	

審査請求 未請求 請求項の数9 O L (全 19 頁)

(21) 出願番号 特願平8-233264

(22) 出願日 平成8年(1996) 9月3日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 ▲高▼橋 正人

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 宮田 裕行

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

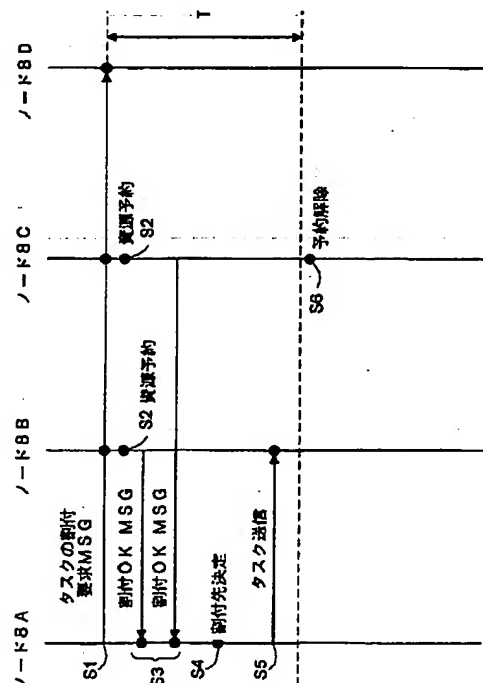
(74) 代理人 弁理士 宮田 金雄 (外3名)

(54) 【発明の名称】 動的負荷分散方法並びにその情報処理装置

(57) 【要約】

【課題】 ネットワークトラヒックを減少させ、高速な処理を可能とすること

【解決手段】 複数の情報処理装置のそれぞれにタスクの実行に必要な資源を予約する資源予約ステップS2と、この資源予約ステップの後に実行され、上記複数の情報処理装置のうちから上記タスクを割り当てる情報処理装置を決定する決定ステップS4と、上記タスクを有する送信側情報処理装置が上記決定ステップが決定した情報処理装置へ上記タスクを送信するタスク送信ステップS5と、上記複数の情報処理装置のそれぞれが上記予約の有効期限を検出し、上記有効期限を過ぎた上記予約を持つ情報処理装置が、自己の資源に対する上記予約を解除する予約解除ステップS6とを備えたものである。



【特許請求の範囲】

【請求項1】 複数の情報処理装置のそれぞれにタスクの実行に必要な資源を予約する資源予約ステップと、この資源予約ステップの後に実行され、上記複数の情報処理装置のうちから上記タスクを割り当てる情報処理装置を決定する決定ステップと、上記タスクを有する送信側情報処理装置が上記決定ステップが決定した情報処理装置へ上記タスクを送信するタスク送信ステップと、

上記複数の情報処理装置のそれぞれが上記予約の有効期限を検出し、上記有効期限を過ぎた上記予約を持つ情報処理装置が、自己の資源に対する上記予約を解除する予約解除ステップと、を備えた動的負荷分散方法。

【請求項2】 タスク送信ステップは、送信側情報処理装置が予約の有効期限が過ぎるまでに、決定ステップが決定した情報処理装置へタスクを送信し、予約解除ステップは、複数の情報処理装置のそれぞれが上記予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに上記タスクが送信されなかった情報処理装置が、自己の資源に対する上記予約を解除することを特徴とする請求項1に記載の動的負荷分散方法。

【請求項3】 予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでにタスクが送信されなかった情報処理装置が、自己の資源に対する上記予約を段階的に解除し、タスク送信ステップで送信されたタスクを受け取った情報処理装置が、上記予約解除ステップが解除した自己の資源に対する予約を回復する予約回復ステップを備えたことを特徴とする請求項2に記載の動的負荷分散方法。

【請求項4】 決定ステップが決定した情報処理装置へ予約確定信号を送信する確定信号送信ステップを備え、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに上記予約確定信号を受信しなかった情報処理装置が、自己の資源に対する上記予約を解除することを特徴とする請求項1に記載の動的負荷分散方法。

【請求項5】 予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに予約確定信号を受信しなかった情報処理装置が、自己の資源に対する上記予約を段階的に解除し、上記予約確定信号を受け取った情報処理装置が、上記予約解除ステップが解除した自己の資源に対する予約を回復する予約回復ステップを備えたことを特徴とする請求項4に記載の動的負荷分散方法。

【請求項6】 複数の第1の情報処理装置それぞれにタスクの実行に必要な資源を第1の資源予約として予約す

る第1の資源予約ステップと、

上記複数の第1の情報処理装置それぞれについて第1の資源情報を調査する第1の資源調査ステップと、

この第1の資源調査ステップが調査した上記第1の資源情報に基づいて、上記複数の第1の情報処理装置のうちからタスクを割り当てる第1の情報処理装置を割当候補として決定する割当候補決定ステップと、

この割当候補決定ステップ終了後に実行され、上記タスクを有する送信側情報処理装置が上記割当候補へ第1の予約延長信号を送信する第1の予約延長信号送信ステップと、

上記複数の第1の情報処理装置のそれぞれが上記第1の資源予約の有効期限を検出し、上記複数の第1の情報処理装置のうち上記有効期限が過ぎるまでに上記第1の予約延長信号が送信されなかった第1の情報処理装置が、上記第1の資源予約ステップで予約した自己の資源に対する第1の資源予約を解除する第1の予約解除ステップと、

第2の情報処理装置に、上記タスクの実行に必要な資源を第2の資源予約として予約する第2の資源予約ステップと上記第2の情報処理装置の第2の資源情報を調査する第2の資源調査ステップと、

上記割当候補の第1の資源情報及び上記第2の資源情報に基づいて、上記割当候補及び上記第2の情報処理装置のうちからタスクを割り当てる情報処理装置を割当情報処理装置として決定する割当先決定ステップと、

上記送信側情報処理装置が上記割当情報処理装置へ第2の予約延長信号を送信する第2の予約延長信号送信ステップと、

上記第2の情報処理装置が上記第2の資源予約の有効期限を検出し、上記第2の資源予約の有効期限が過ぎるまでに上記第2の予約延長信号が送信されなかった場合に、上記第2の資源予約を解除する第2の予約解除ステップと、

上記割当候補が、上記第1の予約延長信号を受け取ってからの時間が予め定められた時間を経過するまでに、上記第2の予約延長信号を受信しなかった場合に、上記第1の資源予約ステップで予約した自己の資源に対する第1の資源予約を解除する第3の予約解除ステップと、

上記第2の予約延長信号送信ステップ終了後に実行され、上記送信側情報処理装置が上記割当情報処理装置へ上記タスクを送信する送信ステップと、を備えた動的負荷分散方法。

【請求項7】 タスクの実行を要求する他の情報処理装置から自己の資源に対する予約を受け付けるとともに、受け付けた予約について自己の資源を予約する予約手段と、

上記予約の有効期限が過ぎるまでに上記タスクを受け取らなかった場合に、上記予約を解除する予約解除手段と、

上記他の情報処理装置から上記タスクを受け取った場合に、上記タスクに上記予約手段で予約した資源を割り当てる資源割当手段と、を備えた情報処理装置。

【請求項 8】 予約解除手段は予約の有効期限が過ぎた場合に上記予約を段階的に解除し、

他の情報処理装置からタスクを受け取った場合に、上記予約解除手段が解除した資源の予約を回復する予約回復手段を備えたことを特徴とする請求項 7 に記載の情報処理装置。

【請求項 9】 他の情報処理装置から受け付けた予約の数に基づいて、予約受付可能な資源量のしきい値を変化させるしきい値変化手段と、
新たな予約の要求があったとき、上記新たな予約及び受け付けた予約の資源量が上記しきい値を超えない場合に、上記新たな予約について自己の資源を予約する予約手段と、
上記新たな予約の有効期限を過ぎた場合に、上記新たな予約を解除する予約解除手段と、
上記新たな予約を要求した情報処理装置からタスクを受け取った場合に、上記タスクに上記予約手段が予約した資源を割り当てる資源割当手段と、を備えた情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、高速処理が要求される並列計算機の動的負荷分散方法並びにその情報処理装置に関するものである。

【0002】

【従来の技術】図 15 は、従来の技術の並列計算機システムを示すものである。図 15 において、100 はネットワークであり、101 はネットワーク 100 に接続されたタスクを処理する処理ノードを表している。

【0003】従来、このような並列計算機システムの動的負荷分散は、文献「Stankovic, J. A., Ramaritham, K., Tutorial: Hard Realtime Systems, IEEE, IEEE Catalog Number EH0276-6, ISBN 0-8186-0819-6, 1988」でも示されるように、各処理ノード 101 は自身の負荷が重い状態でタスクの生成要求が生じた場合、ネットワークを経由して他の処理ノード 101 の余剰計算時間などの資源量を調べる。これにより該処理ノード 101 での処理可否の判定を行った結果、可能であると判定された処理ノード 101 が 1 つ以上あった場合、その内 1 つの処理ノード 101 に対して該タスクの割り当てを依頼することを決定し、該タスクをネットワーク 100 を経由して割り付ける。

【0004】また、より高速な処理が可能な並列計算機の動的負荷分散方法として、本願発明と同一出願人による特願平 7-315394 号がある。この動的負荷分散方法は、タスクを他のノードに処理させるために、タスクの実行に必要な資源を複数のノードそれぞれについて

予め予約する。次に、それらのノードからタスクを割り当てるノードを決定し、そのノードへタスクを送信する。一方、タスクを送信しなかった他のノードへは、予約解除指示を送信する。

【0005】

【発明が解決しようとする課題】このような従来の動的負荷分散システムにおいては、処理可否の判定、タスク割り当ての決定を経て実際にタスクが割り当てられるまでには、ネットワーク 100 の伝達遅延を含めた遅延が発生する。タスクは遅延して決定した処理ノード 101 に到着するが、その時点の資源状況は判定時の資源状況とは変化していることがある。すなわち、並行で進行している他のタスク割り付けが行われることによって、判定時には使用可能であった資源が送信時には使用不能になっていることがあった。この場合、送信されたタスクは該処理ノード 101 において拒否される。その際には、該タスクを割り付け可能な他の処理ノード 101 を再度探さなければならない。また、送信されたタスクは無駄になり、ネットワーク上のトラフィックを増大させる原因ともなる。

【0006】また、上述のようなタスクの再送を防止するために資源を予め予約する動的負荷分散方法では、使用されない資源の予約を解除するために、予約解除指示メッセージを送信することが必要となる。

【0007】以上のように従来の並列計算機システムにおいては、伝達遅延時間のため、決定された処理ノードにおいて、送信されたタスクの処理が拒否されることがあった。そのため、他の処理ノードを探して再送しなければならずネットワークトラフィックの増加を招くという問題があった。また、資源予約を解除するためにメッセージを送信しなければならず、この予約解除メッセージを送信しているため、より高速な処理な処理に適さないという問題があった。

【0008】この発明は、かかる問題点を解決するためになされたもので、ネットワーク上のトラフィックを減少させ、高速な処理が可能な動的負荷分散方法並びにその情報処理装置を得ることを目的としている。

【0009】

【課題を解決するための手段】この発明にかかる動的負荷分散方法においては、複数の情報処理装置のそれぞれにタスクの実行に必要な資源を予約する資源予約ステップと、この資源予約ステップの後に実行され、上記複数の情報処理装置のうちから上記タスクを割り当てる情報処理装置を決定する決定ステップと、上記タスクを有する送信側情報処理装置が上記決定ステップが決定した情報処理装置へ上記タスクを送信するタスク送信ステップと、上記複数の情報処理装置のそれぞれが上記予約の有効期限を検出し、上記有効期限を過ぎた上記予約を持つ情報処理装置が、自己の資源に対する上記予約を解除する予約解除ステップと、を備えたものである。

【0010】また、タスク送信ステップは、送信側情報処理装置が予約の有効期限が過ぎるまでに、決定ステップが決定した情報処理装置へタスクを送信し、予約解除ステップは、複数の情報処理装置のそれぞれが上記予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに上記タスクが送信されなかった情報処理装置が、自己の資源に対する上記予約を解除することを特徴とするものである。

【0011】また、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでにタスクが送信されなかった情報処理装置が、自己の資源に対する上記予約を段階的に解除し、タスク送信ステップで送信されたタスクを受け取った情報処理装置が、上記予約解除ステップが解除した自己の資源に対する予約を回復する予約回復ステップを備えたものである。

【0012】また、決定ステップが決定した情報処理装置へ予約確定信号を送信する確定信号送信ステップを備え、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに上記予約確定信号を受信しなかった情報処理装置が、自己の資源に対する上記予約を解除することを特徴とするものである。

【0013】また、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに予約確定信号を受信しなかった情報処理装置が、自己の資源に対する上記予約を段階的に解除し、上記予約確定信号を受け取った情報処理装置が、上記予約解除ステップが解除した自己の資源に対する予約を回復する予約回復ステップを備えたものである。

【0014】また、複数の第1の情報処理装置それぞれにタスクの実行に必要な資源を第1の資源予約として予約する第1の資源予約ステップと、上記複数の第1の情報処理装置それぞれについて第1の資源情報を調査する第1の資源調査ステップと、この第1の資源調査ステップが調査した上記第1の資源情報に基づいて、上記複数の第1の情報処理装置のうちからタスクを割り当てる第1の情報処理装置を割当候補として決定する割当候補決定ステップと、この割当候補決定ステップ終了後に実行され、上記タスクを有する送信側情報処理装置が上記割当候補へ第1の予約延長信号を送信する第1の予約延長信号送信ステップと、上記複数の第1の情報処理装置のそれぞれが上記第1の資源予約の有効期限を検出し、上記複数の第1の情報処理装置のうち上記有効期限が過ぎるまでに上記第1の予約延長信号が送信されなかった第1の情報処理装置が、上記第1の資源予約ステップで予約した自己の資源に対する第1の資源予約を解除する第1の予約解除ステップと、第2の情報処理装置に、上記タスクの実行に必要な資源を第2の資源予約として予約

する第2の資源予約ステップと、上記第2の情報処理装置の第2の資源情報を調査する第2の資源調査ステップと、上記割当候補の第1の資源情報及び上記第2の資源情報に基づいて、上記割当候補及び上記第2の情報処理装置のうちからタスクを割り当てる情報処理装置を割当情報処理装置として決定する割当先決定ステップと、上記送信側情報処理装置が上記割当情報処理装置へ第2の予約延長信号を送信する第2の予約延長信号送信ステップと、上記第2の情報処理装置が上記第2の資源予約の有効期限を検出し、上記第2の資源予約の有効期限が過ぎるまでに上記第2の予約延長信号が送信されなかった場合に、上記第2の資源予約を解除する第2の予約解除ステップと、上記割当候補が、上記第1の予約延長信号を受け取ってから時間が予め定められた時間を経過するまでに、上記第2の予約延長信号を受信しなかった場合に、上記第1の資源予約ステップで予約した自己の資源に対する第1の資源予約を解除する第3の予約解除ステップと、上記第2の予約延長信号送信ステップ終了後に実行され、上記送信側情報処理装置が上記割当情報処理装置へ上記タスクを送信する送信ステップと、を備えたものである。

【0015】また、この発明にかかる情報処理装置においては、タスクの実行を要求する他の情報処理装置から自己の資源に対する予約を受け付けるとともに、受け付けた予約について自己の資源を予約する予約手段と、上記予約の有効期限が過ぎるまでに上記タスクを受け取らなかった場合に、上記予約を解除する予約解除手段と、上記他の情報処理装置から上記タスクを受け取った場合に、上記タスクに上記予約手段で予約した資源を割り当てる資源割当手段と、を備えたものである。

【0016】また、予約解除手段は予約の有効期限が過ぎた場合に上記予約を段階的に解除し、他の情報処理装置からタスクを受け取った場合に、上記予約解除手段が解除した資源の予約を回復する予約回復手段を備えたものである。

【0017】また、他の情報処理装置から受け付けた予約の数に基づいて、予約受付可能な資源量のしきい値を変化させるしきい値変化手段と、新たな予約の要求があったとき、上記新たな予約及び受け付けた予約の資源量が上記しきい値を超えない場合に、上記新たな予約について自己の資源を予約する予約手段と、上記新たな予約の有効期限を過ぎた場合に、上記新たな予約を解除する予約解除手段と、上記新たな予約を要求した情報処理装置からタスクを受け取った場合に、上記タスクに上記予約手段が予約した資源を割り当てる資源割当手段と、を備えたものである。

【0018】

【発明の実施の形態】

実施の形態1. 図1は、実施の形態1～9に共通のプラットフォームである並列計算機システムを示すものであ

る。図1において、1A~Dはそれぞれタスクを処理するプロセッサ、2A~Dはそれぞれタスクを処理するために必要なプログラム及びデータを記憶するメモリ、3A~Dはそれぞれプロセッサ1A~D、メモリ2A~D、EDU4A~D、及びタイマ7A~Dを接続し、入出力を制御するインタフェースである。4A~Dはそれぞれインタフェース3A~Dとネットワーク5を接続するEDU (External Distribution Unit) であり、インタフェース3A~Dからのデータをネットワーク5上へ送信し、ネットワーク5上を流れる自己の情報処理装置宛てのデータ若しくは命令を受信して、インタフェース3A~Dへ出力する。6A~Dはそれぞれメモリ2A~D内のそれぞれに設けられ、自己のノード8A~Dの総資源量、使用中の資源量、空き資源量、予約資源量の情報が記憶されるテーブルである。7A~Dはそれぞれは、時刻を計測するタイマである。

【0019】8A~Dはそれぞれプロセッサ1A~D、メモリ2A~D、インタフェース3A~D、EDU4A~D、及び、タイマ7A~Dを有する情報処理装置たるノードであり、これらのノード8A~Dはそれぞれ単独でタスクを処理する能力を持っている。なお、ここでは各情報処理装置を接続する信号線として、ネットワーク5が使用されているが、情報を伝達できるものであればよく、例えば、バス等を使用してもよい。

【0020】例えば、ノード8Aでタスクが発生したとき、このタスクをいずれかのノード8 (A~D) に割り当てる。以下に、この場合の動的負荷分散処理を図1~図4を用いて説明する。

【0021】図2はこの実施の形態1の動的負荷分散処理の流れを説明するシーケンス図である。図2において、図1と同一の符号は同一又は相当の部分を表す。図3はこの実施の形態1のタスク送信側ノードのタスク送信処理を説明するフローチャートである。図3において、図2と同一の符号は同一又は相当の処理を表す。図4はこの実施の形態1のタスク受信側ノードのタスク受信処理を説明するフローチャートである。図4において、図2と同一の符号は同一又は相当の処理を表す。

【0022】まず初めに、ノード8Aにおいてタスクが生成され、このタスクを自ノード8Aで処理するか、他ノード8B~Dのいずれかで処理するかを決定する。この処理を図2、図3を用いて説明する。

【0023】まず、ステップS1でタスク送信側のノード8Aが、他のノード8B~Dへタスクの割付要求メッセージを送信する。このメッセージはプロセッサ1AがEDU4Aを介して、ネットワーク5上に放送する。この放送によって、一度に複数のノード8B~Dが割付要求メッセージを受信する。受信はEDU4B~Dによってそれぞれ行われ、プロセッサ1B~Dが他のタスクを処理している間でも正常に行われる。割付要求メッセージには、タスクの実行に必要な資源の情報 (以下、資源

情報という) が含まれている。他のノード8B~Dのそれぞれは、受信した割付要求メッセージの資源情報に基づいて、タスクを受け付けられるか否かを判断する。

【0024】次にステップS2に移り、上述ステップS1でタスクを受け付けられると判断した場合には、資源情報に記載された資源を予約し、他のタスクがその資源を使用できないようにする。受け付けられないと判断した場合には、資源の予約は行わない。

【0025】続いて、ステップS3で、タスクを受け付けられると判断したノード8B、Cからタスク送信側ノード8Aへ、割付を受諾する旨の割付OKメッセージを送信する。ここでは、受け付けられないと判断したノード8Dは、ネットワークトラフィックを少なくするためメッセージを送信しない。送信側ノード8Aは、予め定められた時間まで割付OKメッセージを受け付け、次のステップS4に移る。

【0026】ステップS4では、送信側ノード8Aが割付OKメッセージを送信してきたノード8B、Cのうちからタスクを割り付けるノードを決定する。この決定は、ランダムにノードを選択してもよいし、最も資源に余裕のあるノードを選択して行ってもよい。このステップS4でタスク割付先として決定されたノード8Bは、割付情報処理装置となる。

【0027】次に、ステップS5で送信側ノード8Aがタスク本体をネットワーク5上へ送信する。送信は、割付情報処理装置たるノード8Bを送信先として指定して行う。ノード8Bはネットワーク5を介してタスクを受け取り、ステップS2で予約していた自己の資源を受け取ったタスクに割り当てる。そして、ノード8Bにて、タスクが実行される。

【0028】一方、タスクを受信しなかったノード8CはステップS6として、予約の有効期限を過ぎていないかを判断し、過ぎていた場合には予約を解除する。ここでは、タスクの割付要求メッセージを受信してから予め定められた時間T (以下、予約解除時間Tという) 経過後に、ステップS2で行った資源の予約を解除する。この予約解除時間Tの計測はタイマ7Cを用いて行う。予約が解除されたことにより、他のタスクはその資源を使用できるようになる。予約の有効期限は、タスクの割付要求メッセージの受信時に基づいた期限であればよく、資源を予約してから予め定められた時間で設定してもよいし、割付OKメッセージを送信してから予め定められた時間を設定してもよい。

【0029】このとき、予約解除時間Tが経過すると自動的に予約が解除されるため、ネットワーク5を介した予約解除メッセージの送信は行われない。従って、ネットワーク5上を送受信するメッセージ数を減らすことができる。すなわち、ネットワークトラフィックを減少させることができる。また、動的負荷分散を行う並列処理装置では、ノード8A~D間のデータ授受が頻繁に行わ

10

20

30

40

50

れ、多くのノード8A~Dがネットワーク5を使用するため、ネットワーク5のトラヒックが増大するとデータの送受信待ちを生ずる時間が長くなり、処理速度が低下する。逆に、ネットワーク5のトラヒックを少なく抑えることによって並列処理装置全体の処理効率も向上し、高速な処理が可能となる。

【0030】◆タスク送信側ノードの処理

次に上述の動的負荷分散処理をタスク送信側ノード8Aの処理に絞って説明する。初めに、図3のステップSステップ101において、ノード8Aがタスクを生成する。このタスクはノードの外部から受け取ったものであってもよい。次に、ステップS102に移り、タスクが自ノードで処理可能かを判断する。処理可能であるか否かを判断する方法は、例えば、余剰計算時間、空きメモリがタスクを実行するのに十分であるかを判断することによって行う。ここで、自ノードで処理できると判断した場合には、ステップS103に移り、処理できないと判断した場合にはステップS1-1に移る。

【0031】ステップS103では、タスクを自ノードで実行する。実行が終了すると、また、ステップS101に戻る。ただし、このノードでは、複数のプロセス、スレッドを並行処理することができる。そのため、ステップS103の処理が終了しなくとも、ステップS101の処理が行われ、次々にタスクが生成される。

【0032】一方、上述ステップS102にて他ノード8B~Dにタスクを処理させることが決定すると、図2のステップS1で示すようにタスクの割付要求メッセージの送受信が行われる。タスク送信側のノード8Aでは、図3のステップS1-1が実行され、他のノード8B~Dへタスク割付要求信号を送信する。この割付要求メッセージは、図5のM1に示すようにメッセージのID、メッセージ種別、発信元、発信先、必要資源等のデータによって構成されている。ここで、メッセージ種別には「割付要求」を示す情報が入り、発信元はタスク送信側のノード8A、発信先はノード8B~Dへの同報通信である旨の情報が記載されている。データ部分には、タスクを識別するタスク番号とタスクの実行に必要な計算時間及びメモリ等の必要資源情報が記載されている。

【0033】次にステップ3-2において、割付OKメッセージを予め定められた時間の間受信待ちする。この受信待ち時間は、タスク、又はノード数等の条件に応じて変化するようにしてもよい。割付OKメッセージは図5のM2に示すように、タスク番号を含む。このタスク番号を識別することにより、その割付OKメッセージが当該タスクに対するものであるかを判別することができる。

【0034】続いて、ステップS4で、割付OKメッセージを送信してきたノード8B、Cのうちから、タスクを送信するノード8を割付情報処理装置（以下、割当ノードという）として決定する。次に、ステップS5-1

に移り、割当ノードへタスク送信メッセージを送信する。このタスク送信メッセージは、図5のM3に示したような構造を有し、発信先として割当ノードの識別子が指定され、データ部にはタスク本体が含まれている。

【0035】以上で動的負荷分散処理が終了する。そして、また新たなタスクがステップS101で生成され、同様の処理が行われる。このタスク送信側ノード8Aの処理では、割付ノード以外へ予約解除メッセージを送信しないため、タスク送信側ノード8Aの処理負担が軽いという特徴がある。

【0036】◆他のノード（タスク受信側ノード）の処理

次に上述の動的負荷分散処理をタスク受信側のノード8B~Dの処理に絞って説明する。

【0037】まず、図4のステップS1-2でタスクの割付要求メッセージを受信し、要求を受け付けることができるかどうかを判断する。受け付けできないと判断した場合には、以降の処理を行わず、新たな要求が到着するのを待つ。受け付けできると判断した場合には、次のステップS1-3に移る。ステップS1-3では、タイマ7B~Dをそれぞれ用いて時間tの計測を開始する。すなわち、時間長の計測を開始する。このステップS1-3は次のステップS2の後に実行してもよい。続いて、ステップS2でタスクの実行に必要な資源を予約する。予約する資源は、タスクの割付要求メッセージに含まれている資源情報から必要な資源の量を決定し、この決定した量の分だけ自己のノードの管理する資源を予約する。

【0038】続いて、ステップS3-1に移り、割付OKメッセージを送信する。次にステップS5-2に移り、EDU4B~Dがタスクを受信したかを調べる。受信した場合は、ステップS5-4に移り、ステップS2で予約した資源を受信したタスクに割り当てる。続いて、ステップS111でタスクを実行する。

【0039】一方、ステップS5-2でタスクを受信していないと判断された場合には、ステップS5-3に移り、ステップS1cから計測を開始している時間tが予約解除時間Tを経過しているかを検出する。時間tが予約解除時間Tを過ぎていない場合には、ステップS5-2に戻る。予約解除時間Tを過ぎている場合には、ステップS6に移る。ステップS6では、ステップS2で行った資源の予約を解除し、当該資源が他のタスクで使えるようにする。

【0040】以上のように、資源の予約を解除するメッセージを受信しなくとも、予約が解除されるため、ネットワークトラヒックを減少させ、システム全体として処理速度が向上するという効果がある。

【0041】実施の形態2。実施の形態1では、タスクの割付要求メッセージを放送（ブロードキャスト）したが、1つ1つのノードに逐次送信してもよい。以下にこ

の実施の形態2の動作について図6を用いて説明する。図6はこの実施の形態1の動的負荷分散処理の流れを説明するシーケンス図である。図6において、図2と同一の符号は同一又は相当の部分を表す。

【0042】まず、ステップS1aにおいて、タスク送信側ノード8Aがタスクの割付要求メッセージを送信する。このタスクの割付要求メッセージには1つのノード8Bが送信先として指定されて送信される。次にステップS2aにて、ノード8Bが図2のステップS2にて説明したのと同様に資源予約を行う。続いてステップS3aにて、ノード8Bからタスク送信側ノード8Aへ、割付OKメッセージが送信される。タスク割付が不可能なときはこの割付OKメッセージは送信されない。

【0043】次に、ステップS1b、S2b、及びS3bにて、タスク送信側ノード8Aが他のノード8Cへタスクの割付要求メッセージを送信し、タスク割付可能かどうかを調査する。この調査は、上述のステップS1a、S2a、及びS3aと同様に行われる。同様に、他のノード8Dについてもタスク割付可能かどうかを調査する。この調査は上述のステップS1c、S2c、及びS3cと同様に行われる。

【0044】次に、ステップS4に移り、割付先決定処理を行い、続くステップS5で割付情報処理装置たる割付ノード8Bへタスクを送信する。各ノード8B〜Dそれぞれはタスクの割付要求メッセージを受け取ってからの時間tを計測し、ステップS5のタスク送信は、計測した時間tが予約解除時間Tを経過する前に行う。また、予約解除時間Tまでにタスクが送信されなかったノード8C、Dは、ステップS6にて、資源の予約を解除する。

【0045】この実施の形態2によれば、資源の予約を解除するメッセージを受信しなくとも、予約が解除されるため、ネットワークトラフィックを減少させ、システム全体として処理速度が向上するという効果がある。

【0046】なお、割付先決定ステップS3は、新しい割付OKメッセージが到着する度に行ってもよい。そして、予め定められた資源条件を満たした情報処理装置（すなわちノード）を発見した時点で、他のノードへのタスク割付要求メッセージの送信を行わないにしてもよい。この場合には、タスク要求メッセージと割付OKメッセージによって発生するトラフィックを低く抑えることができるという効果がある。上述の資源条件とは、タスクを余裕を持って実行することのできる資源条件であり、タスクの実行に必要な資源量に基づいて決定される。例えば、当該ノードの余剰計算時間、又はメモリが、タスクの実行に必要な計算時間、又はメモリの20パーセント分存在するという条件である。

【0047】実施の形態3. 実施の形態3は、予約を確定する予約確定メッセージを送信することにより、予約解除時間Tまでにタスクが送信されない場合でも予約が

解除されないようにする実施の形態である。

【0048】図7はこの実施の形態3の動的負荷分散の処理を示すシーケンス図である。図7において、図2と同一の符号は同一又は相当の部分を表す。

【0049】この実施の形態3の基本的な動作は実施の形態1と同様であるため、この実施の形態3における特徴的な動作を以下に説明する。ステップS1〜ステップS4までの処理が終了し、割付ノードが決定すると、ステップS41で、タスク送信側ノード8Aが割付ノードへ予約確定メッセージを送信する。この予約確定メッセージを受け取ったノード8Bは、ステップS2で行われた資源の予約を確定し、ステップS6で予約解除が行われないようにする。

【0050】予約確定メッセージが送信されなかったノード8C、Dでは、ステップS6で予約解除処理が行われる。

【0051】ステップS41の終了後、ステップS5において、タスク送信が行われる。割付ノード8Bでは、ステップS41により予約確定しておいた資源を受信したタスクに割り付ける。

【0052】以上のように、この実施の形態3では、予約解除時間T内にタスクを送信できない場合であっても、ネットワークトラフィックを減少させ、システム全体として処理速度が向上するという効果がある。

【0053】なお、この実施の形態3の予約確定メッセージは、実施の形態2にも適用できる。図8はこの場合の動的負荷分散の処理を説明するシーケンス図である。図8において、図6と同一の符号は同一又は相当の部分を表す。ステップS41は上述の図7のステップS41と同様であり、この予約確定メッセージにより、上述の効果をj得ることができる。

【0054】実施の形態4. 実施の形態4は、良好な割付ノードが見つかるまで、現在の割付ノード候補の予約を延長する実施の形態である。

【0055】図9はこの実施の形態4の動的負荷分散の処理を示すシーケンス図である。図9において、図6と同一の符号は同一又は相当の部分を表す。

【0056】この実施の形態4の基本的な動作は実施の形態2と同様であるため、この実施の形態4における特徴的な動作を以下に説明する。ステップS1a〜S3bまでにおいて、第1の資源調査ステップとして、第1の情報処理装置たる複数のノード8B、Cの資源状況を調査し、第1の資源予約ステップを実行する。次に、ステップS4aに移り、割当候補決定ステップたる割付先決定処理を行う。この割付先決定処理にて、割付OKメッセージを送信してきたノード8B、Cのうちから、最も資源状況の優良なノードを割当候補（以下割当ノード候補という）として決定する。資源状況が優良か否かは、例えば、余剰資源が多くあるか否かで決定する方法や、タスクの実行が他のノードよりも高速に実行できる等の

条件をみて決定する方法がある。

【0057】続いて、予め定められた予約解除時間T1が経過する前に、ステップS42aにて第1の予約延長信号たる予約延長メッセージを割当ノード候補へ送信する。この割当ノード候補8Bは、予約延長メッセージを受け取ると、ステップS1aから計測していた時間をクリアし、新たに時間計測を開始する。

【0058】次に、予約延長メッセージ送信後に、第2の情報処理装置となるノード、すなわち、新たにタスクを割付可能なノードを探すため、第2の資源調査ステップたるステップS1c、S3c、及び第2の資源予約ステップたる資源予約S2cを実行する。次に、割当先決定ステップたるステップS4bに移り、予約延長メッセージを送信してから新たに発見した割当可能なノード8Dの資源状況と、ステップS4aで決定した割当ノード候補8Bの資源状況とを比較し、これらのノード8C、Dのうちから資源状況が優良なノードを探して割当ノード候補8Dを新たに決定する。

【0059】タスク送信側ノード8Aは、割当ノード候補8Bに予約延長メッセージが到着してからの時間が、予め定められた予約解除時間T2を経過する前に、第2の予約延長信号たる予約延長メッセージの送信を第2の予約延長信号送信ステップとして実行する。以上のように、次々と他のノードの資源状況を調査しながら、割当ノードの予約を延長する。この割当ノードの探索と予約の延長は時間の許す限り繰り返し行われ、最適な割当ノードを探し出す。

【0060】ここでは、一回の予約延長後に時間がなくなり、最後の割付先決定ステップS4bで決定した割付ノード候補8Dを割付ノードとして決定したとする。この場合においては、上述の予約延長メッセージの代わりにステップS5でタスクの送信が行われる。タスク送信は、タスク送信側ノード8Aから割付先決定ステップS4bで決定した割付ノード8Dへタスクを送信する処理である。

【0061】一方、予約解除時間T1を過ぎた資源予約は、第1の予約解除ステップたるステップS6bにおいて解除される。同様に、予約解除時間T2を過ぎた資源予約は、第3の予約解除ステップたるステップS6aにおいて解除される。以上は、第2の情報処理装置たる割付ノード8Dへタスクを送信した場合を説明したが、割当候補ノード8Bへタスクを送信した場合には、第2の予約解除ステップとして、第2の情報処理装置たる割付ノード8Dで、ステップS6bと同様に予約が解除される。これにより、上述の実施の形態と同様に予約解除メッセージが不要になる。

【0062】以上のように、この実施の形態4によれば、資源の予約を解除するメッセージを受信しなくとも、予約が解除されるため、ネットワークトラフィックを減少させ、システム全体として処理速度が向上するとい

う効果がある。

【0063】また、実施の形態1では、予約解除時間Tとして多くのノードから割付OKメッセージを受け取るに十分な時間を設定する必要があったが、この実施の形態4では予約解除時間T1若しくはT2をそれよりも少なく設定することができる。その理由は、例えば、T1では少なくとも2つのノードから割付OKメッセージを受け取るに十分な時間を設定すればよく、T2では少なくとも1つのノードから割付OKメッセージを受け取るに十分な時間を設定すればよいからである。ただし、予約延長メッセージはトラフィックを増加させる原因となるため、その点において、一回の予約延長で調査できるノード数及び予約解除時間T1、T2の長さは調整される。なお、T1、T2は同じ時間を設定してもよい。

【0064】上述のように、予約解除時間T1、T2を短く設定することができるため、システム全体として、予約されている資源の量を削減することができ、多くのタスクを実行することができるという効果がある。

【0065】なお、タスクの割付要求メッセージは、1つ1つのノード8B～Dに対して個別に送信するのではなく、図10に示すように1つの割付要求メッセージが複数のノードに受信されるようにしてもよい。図10において、図9と同一の符号は同一又は相当の部分を表している。ステップS1のタスクの割付要求メッセージは、図2に示したステップS1のタスクの割付要求メッセージと同様のものである。従って、ステップS1のタスクの割付要求メッセージ1つで、図9のステップS1a、S1b、S1cのタスクの要求メッセージの役割を果たしている。

【0066】実施の形態5。実施の形態5は、予約した資源を段階的に削除する実施の形態である。図11はこの実施の形態5の動的負荷分散の処理を示すシーケンス図である。図11において、図2と同一の符号は同一又は相当の部分を表す。

【0067】この実施の形態5の基本的な動作は実施の形態1と同様であるため、この実施の形態5における特徴的な動作を以下に説明する。ステップS1～S4までにおいて、複数のノード8B～Dのうちから割付ノードを決定する。次に、タスクの送信を実行するのであるが、ここではタスク送信ステップS5が、タスクの送信準備で予約解除時間Tまでに送信できなかったとする。

【0068】予約解除時間Tまでにタスクを受信しなかったノード8B～Dのそれぞれは、ステップS9aにて、ステップS2で行った資源予約の一部を解除する。この場合は予約した資源量の20パーセントを解除する。次に、ステップS9aから予め定められた時間経過後に、ノード8B～Dのそれぞれにおいて、ステップS9bが実行される。ステップS9bでは、ステップS2で行った資源の予約の一部を解除する。ここでは、予約

した資源量の50パーセントを解除する。続いて、ステップS9bから予め定められた時間経過後に、ノード8B~Dのそれぞれにおいて、ステップS9cが実行される。ステップS9cでは、ステップS2で行った資源の予約の全てを解除する。ただし、予約が確定していたり、タスクが送信されているノードでは、これらのステップS9a~cの予約解除は行わない。

【0069】このステップS9a~cまでにおける段階的な予約の解除は、上述のように3回に分けるのみならず、適宜所定の回数を予め定め、この予め定められた回数に従って段階的に削除することができる。また、一回の削除量も適宜設定される。ここで、「段階的な解除」とは、自己のノードに予約されている資源の予約の一部を徐々に解除して行き、最終的に自己のノードの資源に対する予約の全てを解除することをいう。

【0070】一方、ステップS5が図11に示すようなタイミングで実行されると、タスクを受け取ったノード8Bは、ステップS10にて、削除した予約資源の回復を行ない、タスクの実行に必要な資源量が完全に予約されている状態に戻す。ステップS10終了後は、回復した予約資源が受信したタスクに割り当てられ当該タスクが実行される。

【0071】この実施の形態5では、タスクの送信が予約解除時間Tを過ぎた場合でも、資源を段階的に削除するため、タスクが割り付けられる可能性が高いという効果がある。これは、大きな資源を確保するよりも、小さい資源を確保できる可能性の方が高いからである。

【0072】また、実施の形態3と同様に予約確定メッセージを送信し、予約確定メッセージを受け取ったノードが削除した資源予約を回復するようにしてもよい。図12はこの場合に動的負荷分散処理を説明するシーケンス図である。図12において、図11と同一の符号は同一又は相当の部分を表す。予約解除時間T1を過ぎると、上述のステップS9a~S9cで説明したように、予約資源を段階的に解除する。一方、予約確定メッセージを受信したノードは、予約の段階的解除を行わず、又は段階的解除を中止し、ステップS10の予約回復を行う。

【0073】また、実施の形態4と同様に予約延長メッセージを送信し、予約延長メッセージを受け取ったノードが削除した資源予約を回復するようにしてもよい。図13はこの場合に動的負荷分散処理を説明するシーケンス図である。図13において、図11と同一の符号は同一又は相当の部分を表す。予約解除時間T1を過ぎると、上述のステップS9a~S9cで説明したように、予約資源を段階解除する。一方、予約解除時間T1が過ぎた後に、予約延長メッセージが送信された場合には、新たに予約解除時間T2を設定し、上述の予約回復ステップS10を実行する。予約解除時間T2経過までに予約延長メッセージ若しくはタスク送信が無かった場合に

も同様に、予約資源を段階解除する。

【0074】実施の形態6. 実施の形態6は、ノードの保有する空き資源量を越えて予約を受け付ける実施の形態である。図14はこの実施の形態6のレシーバ側システムのタスク受け付け処理のフローチャートである。このフローチャートは、図4のステップS1-2を詳細に説明したものである。

【0075】まず、ステップS1-21でしきい値Thを算出する。このしきい値Thは、他の情報処理装置から受け付けた予約の数に基づいて設定される。例えば、受け付けている予約の数が1であるときは、空き資源量の100パーセントの資源量、予約の数が5のときは、空き資源量の150パーセント、予約の数が10であるときは、空き資源量の200パーセントがしきい値Thとして設定される。

【0076】次にステップS1-22で、タスクの割付要求メッセージを受信する。続いて、ステップS1-23で、ステップS1-22で受け付けたタスクの割付要求の要求する資源量がしきい値Thよりも大きいかなかを判断する。大きいときはタスク割付不可と判断され、ステップS1-21へ戻る。ステップS1-21へ戻ると最新の空き資源量に基づいて新たなしきい値Thが設定される。一方、小さいときはタスク割付可と判断され、図4のステップS1-3へ移る。

【0077】ここで、資源予約の数に応じてしきい値Thを変化させているため、空き資源量を越えて予約を受け付けたとしても、タスクを受け付けられる可能性が高い。すなわち、資源の予約は時限解除される可能性があり、タスクを受け付けられるノード数が多くなればなるほどその確率は高い。従って、予約されたにもかかわらず使用されない資源が多く発生するため、システム全体で実行できるタスクの数が少なく制限されてしまう。一方、複数の予約を受け付けた場合には、その予約数が増加すればするほど、そのうちのいずれかの予約が解除される可能性が高くなる。以上の理由から、この実施の形態6は、資源予約の数に応じてしきい値Thを変化させて、空き資源量を越えた資源量の予約を受け付け、多くのタスクを実行できるようにしている。

【0078】

【発明の効果】この発明は、以上に説明したように構成されているので、以下に記載されるような効果を奏する。

【0079】この発明にかかる動的負荷分散方法においては、複数の情報処理装置のそれぞれにタスクの実行に必要な資源を予約する資源予約ステップと、この資源予約ステップの後に実行され、上記複数の情報処理装置のうちから上記タスクを割り当てる情報処理装置を決定する決定ステップと、上記タスクを有する送信側情報処理装置が上記決定ステップが決定した情報処理装置へ上記タスクを送信するタスク送信ステップと、上記複数の情

報処理装置のそれぞれが上記予約の有効期限を検出し、上記有効期限を過ぎた上記予約を持つ情報処理装置が、自己の資源に対する上記予約を解除する予約解除ステップと、を備え、予約を時限解除するため、ネットワークトラフィックを減少させ、高速な処理が可能となる。

【0080】また、タスク送信ステップは、送信側情報処理装置が予約の有効期限が過ぎるまでに、決定ステップが決定した情報処理装置へタスクを送信し、予約解除ステップは、複数の情報処理装置のそれぞれが上記予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに上記タスクが送信されなかった情報処理装置が、自己の資源に対する上記予約を解除し、予約を時限解除するため、ネットワークトラフィックを減少させ、高速な処理が可能となる。

【0081】また、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでにタスクが送信されなかった情報処理装置が、自己の資源に対する上記予約を段階的に解除し、タスク送信ステップで送信されたタスクを受け取った情報処理装置が、上記予約解除ステップが解除した自己の資源に対する予約を回復する予約回復ステップを備え、予約を時限解除するため、ネットワークトラフィックを減少させ、高速な処理が可能となるとともに、予め定められた時間を経過した後にタスクが割り当てられた場合であっても資源を回復でき、タスクを実行できる可能性が高い。

【0082】また、決定ステップが決定した情報処理装置へ予約確定信号を送信する確定信号送信ステップを備え、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに上記予約確定信号を受信しなかった情報処理装置が、自己の資源に対する上記予約を解除し、予約を時限解除するため、ネットワークトラフィックを減少させ、高速な処理が可能となるとともに、予約確定を予め知らせることができるため、タスクの送信に時間がかかってしまった場合でも予約が解除されることがない。

【0083】また、予約解除ステップは、複数の情報処理装置のそれぞれが予約の有効期限を検出し、上記複数の情報処理装置のうち上記有効期限が過ぎるまでに予約確定信号を受信しなかった情報処理装置が、自己の資源に対する上記予約を段階的に解除し、上記予約確定信号を受け取った情報処理装置が、上記予約解除ステップが解除した自己の資源に対する予約を回復する予約回復ステップを備え、予約を時限解除するため、ネットワークトラフィックを減少させ、高速な処理が可能となるとともに、予め定められた時間を経過した後にタスクが割り当てられた場合であっても資源を回復できる可能性が高い。

【0084】また、複数の第1の情報処理装置それぞれ

にタスクの実行に必要な資源を第1の資源予約として予約する第1の資源予約ステップと、上記複数の第1の情報処理装置それぞれについて第1の資源情報を調査する第1の資源調査ステップと、この第1の資源調査ステップが調査した上記第1の資源情報に基づいて、上記複数の第1の情報処理装置のうちからタスクを割り当てる第1の情報処理装置を割当候補として決定する割当候補決定ステップと、この割当候補決定ステップ終了後に実行され、上記タスクを有する送信側情報処理装置が上記割当候補へ第1の予約延長信号を送信する第1の予約延長信号送信ステップと、上記複数の第1の情報処理装置のそれぞれが上記第1の資源予約の有効期限を検出し、上記複数の第1の情報処理装置のうち上記有効期限が過ぎるまでに上記第1の予約延長信号が送信されなかった第1の情報処理装置が、上記第1の資源予約ステップで予約した自己の資源に対する第1の資源予約を解除する第1の予約解除ステップと、第2の情報処理装置に、上記タスクの実行に必要な資源を第2の資源予約として予約する第2の資源予約ステップと、上記第2の情報処理装置の第2の資源情報を調査する第2の資源調査ステップと、上記割当候補の第1の資源情報及び上記第2の資源情報に基づいて、上記割当候補及び上記第2の情報処理装置のうちからタスクを割り当てる情報処理装置を割当情報処理装置として決定する割当先決定ステップと、上記送信側情報処理装置が上記割当情報処理装置へ第2の予約延長信号を送信する第2の予約延長信号送信ステップと、上記第2の情報処理装置が上記第2の資源予約の有効期限を検出し、上記第2の資源予約の有効期限が過ぎるまでに上記第2の予約延長信号が送信されなかった場合に、上記第2の資源予約を解除する第2の予約解除ステップと、上記割当候補が、上記第1の予約延長信号を受け取ってから時間が予め定められた時間を経過するまでに、上記第2の予約延長信号を受信しなかった場合に、上記第1の資源予約ステップで予約した自己の資源に対する第1の資源予約を解除する第3の予約解除ステップと、上記第2の予約延長信号送信ステップ終了後に実行され、上記送信側情報処理装置が上記割当情報処理装置へ上記タスクを送信する送信ステップと、を備え、予約を時限解除するため、ネットワークトラフィックを減少させ、高速な処理が可能となるとともに、第1の資源予約と第2の資源予約とに分けて資源を予約し、解除するため、一度に予約する資源の量をシステム全体として少なくすることができる。

【0085】また、この発明にかかる情報処理装置においては、タスクの実行を要求する他の情報処理装置から自己の資源に対する予約を受け付けるとともに、受け付けた予約について自己の資源を予約する予約手段と、上記予約の有効期限が過ぎるまでに上記タスクを受け取らなかった場合に、上記予約を解除する予約解除手段と、上記他の情報処理装置から上記タスクを受け取った場合

に、上記タスクに上記予約手段で予約した資源を割り当てる資源割当手段と、を備え、予約を時限解除するため、ネットワークトラヒックを減少させ、高速な処理が可能となる。

【0086】また、予約解除手段は予約の有効期限が過ぎた場合に上記予約を段階的に解除し、他の情報処理装置からタスクを受け取った場合に、上記予約解除手段が解除した資源の予約を回復する予約回復手段を備え、予約を時限解除するため、ネットワークトラヒックを減少させ、高速な処理が可能となるとともに、予め定められた時間を経過した後にタスクが割り当てられた場合であっても資源を回復できる可能性が高い。

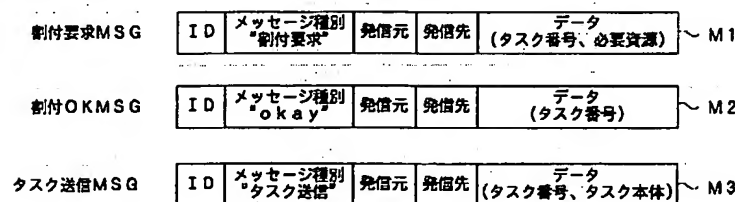
【0087】また、他の情報処理装置から受け付けた予約の数に基づいて、予約受付可能な資源量のしきい値を変化させるしきい値変化手段と、新たな予約の要求があったとき、上記新たな予約及び受け付けた予約の資源量が上記しきい値を超えない場合に、上記新たな予約について自己の資源を予約する予約手段と、上記新たな予約の有効期限を過ぎた場合に、上記新たな予約を解除する予約解除手段と、上記新たな予約を要求した情報処理装置からタスクを受け取った場合に、上記タスクに上記予約手段が予約した資源を割り当てる資源割当手段と、を備え、予約を時限解除するため、ネットワークトラヒックを減少させ、高速な処理が可能となるとともに、予約の数に基づいて予約受付可能な資源量のしきい値を変化されるため、受け付けられる予約の数が増え、かつ送信されたタスクを実行できないという現象を抑制しながら多くのタスクを実行することができる。

【図面の簡単な説明】

【図1】 この発明の実施の形態1における並列計算機システムを示す機能ブロック図である。

【図2】 この発明の実施の形態1における動的負荷分散の処理を示すシーケンス図である。

【図5】



* 【図3】 この発明の実施の形態1におけるセンタ側システムのタスク割り当て処理のフローチャートである。

【図4】 この発明の実施の形態1におけるレシーバ側システムのタスク受け付け処理のフローチャートである。

【図5】 この発明の実施の形態1における各種メッセージの内容を示す図である。

【図6】 この発明の実施の形態2における動的負荷分散の処理を示すシーケンス図である。

【図7】 この発明の実施の形態3における動的負荷分散の処理を示すシーケンス図である。

【図8】 この発明の実施の形態3における他の動的負荷分散の処理を示すシーケンス図である。

【図9】 この発明の実施の形態4における動的負荷分散の処理を示すシーケンス図である。

【図10】 この発明の実施の形態4における他の動的負荷分散処理を示すシーケンス図である。

【図11】 この発明の実施の形態5における動的負荷分散の処理を示すシーケンス図である。

【図12】 この発明の実施の形態5における他の動的負荷分散の処理を示すシーケンス図である。

【図13】 この発明の実施の形態5における他の動的負荷分散の処理を示すシーケンス図である。

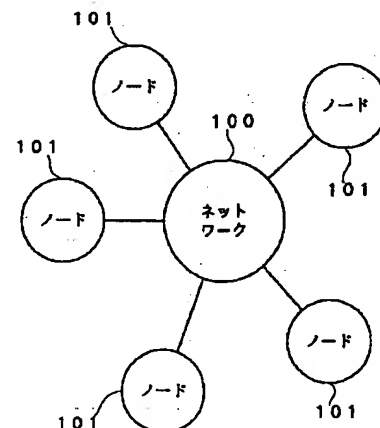
【図14】 この発明の実施の形態6におけるレシーバ側システムのタスク受け付け処理のフローチャートである。

【図15】 従来の技術における並列計算機システムを示す機能ブロック図である。

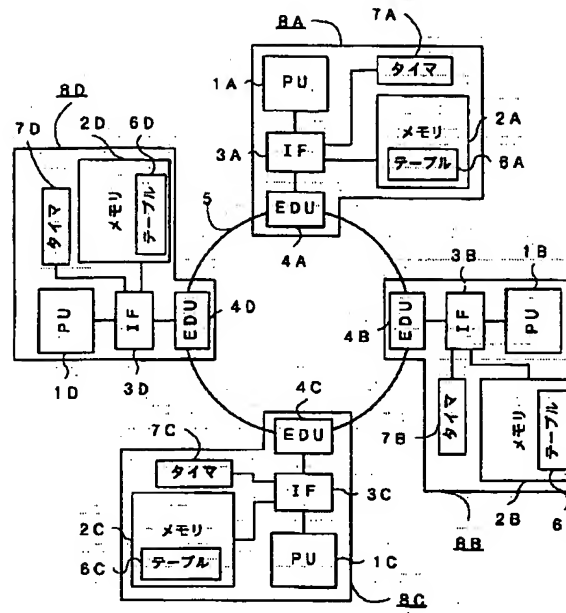
【符号の説明】

1 A～D プロセッサ、 2 A～D メモリ、 3 A～D インタフェース、 4 EDU、 5 ネットワーク、 6 A～D テーブル、 7 A～D タイマ、 8 A～D ノード

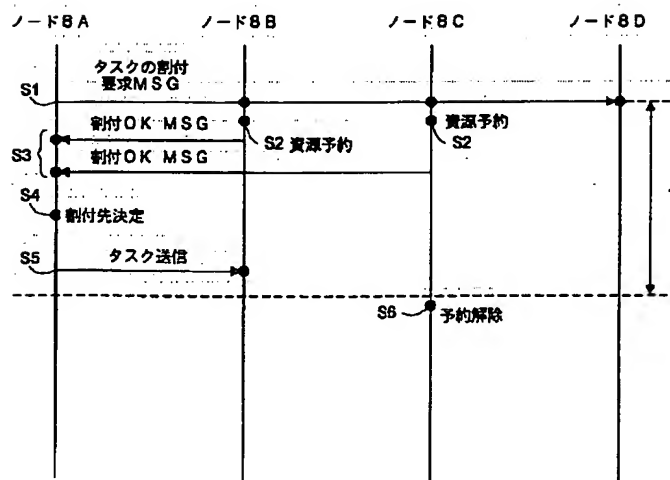
【図15】



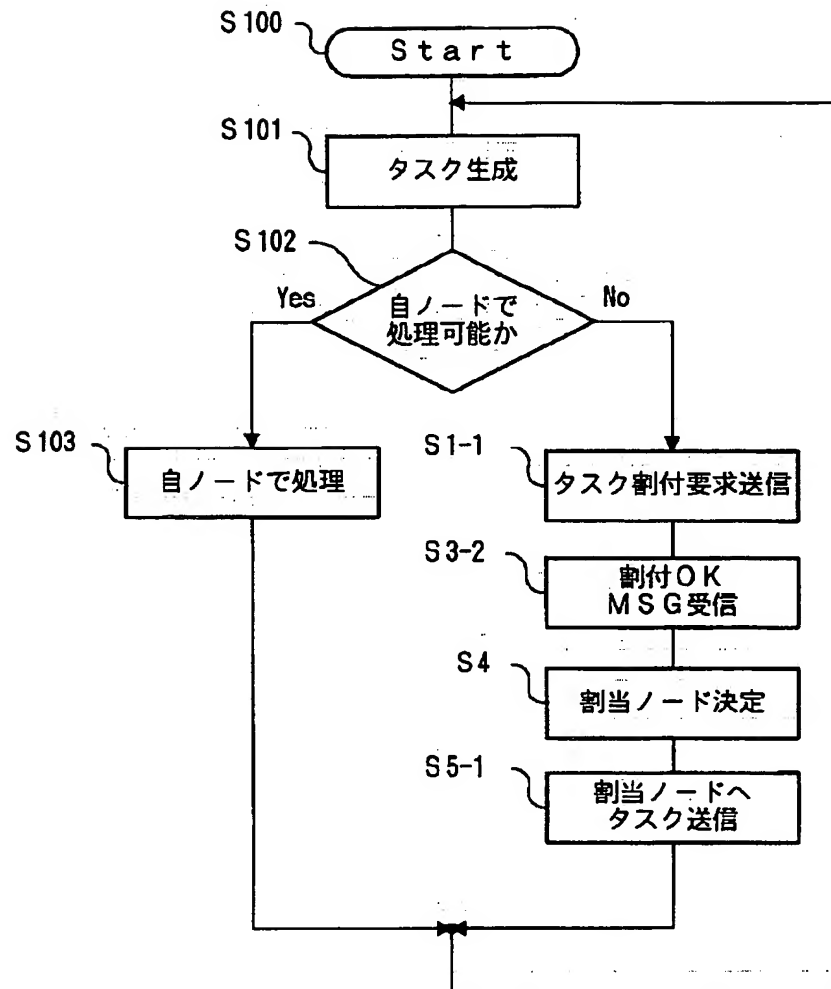
【図1】



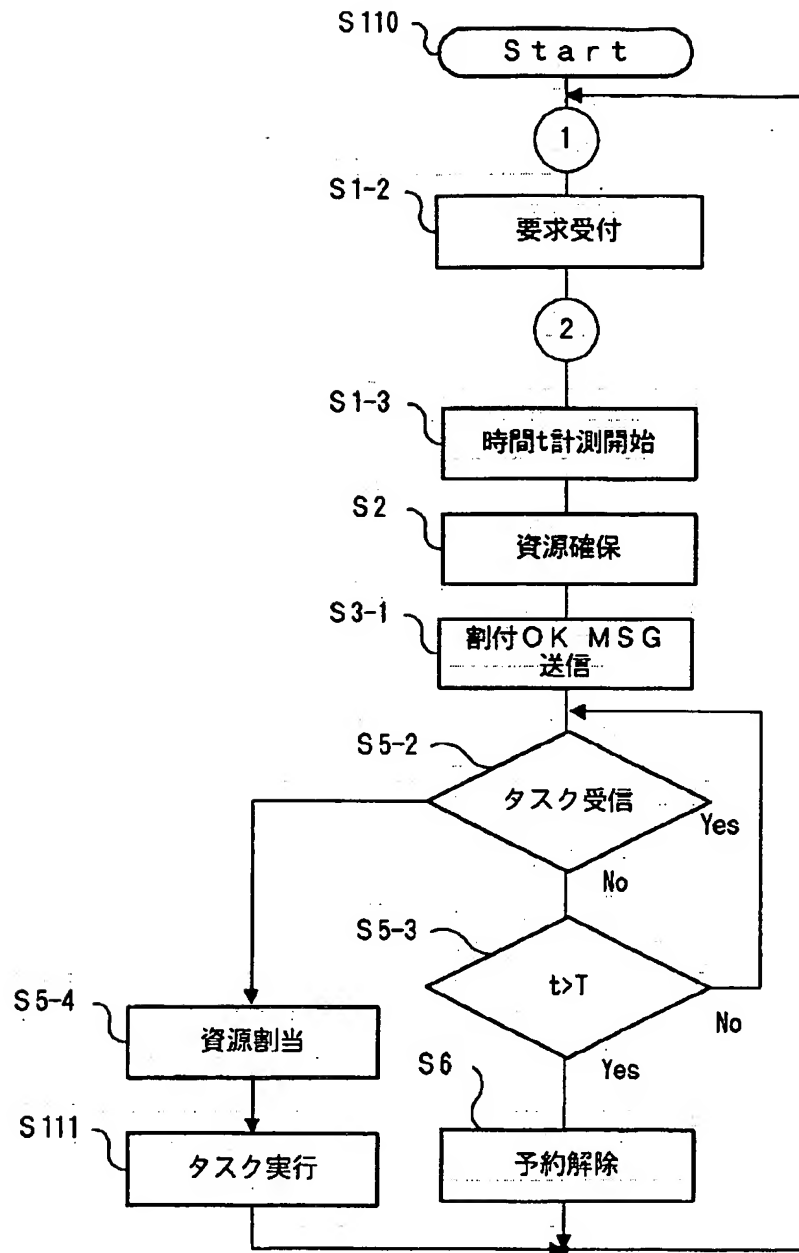
【図2】



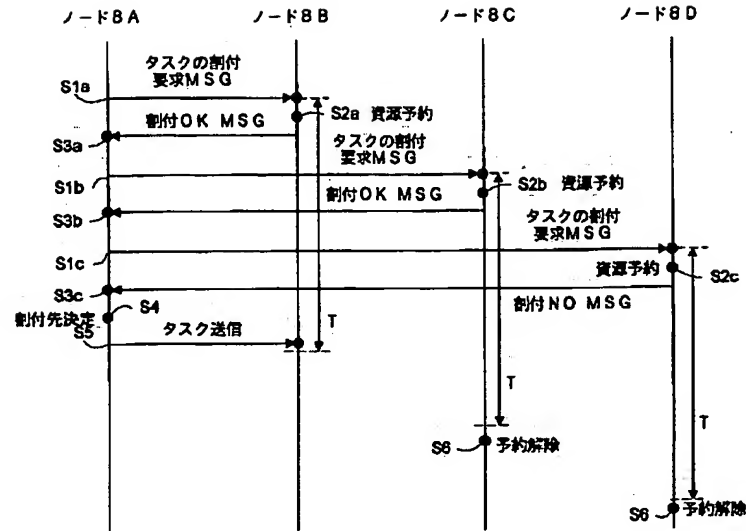
【図3】



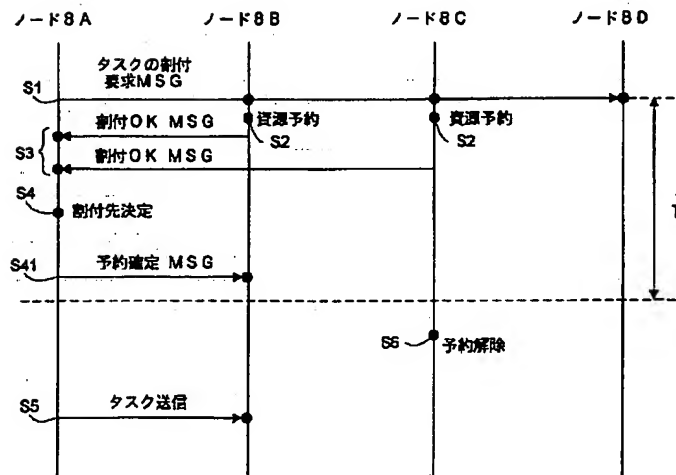
【図4】



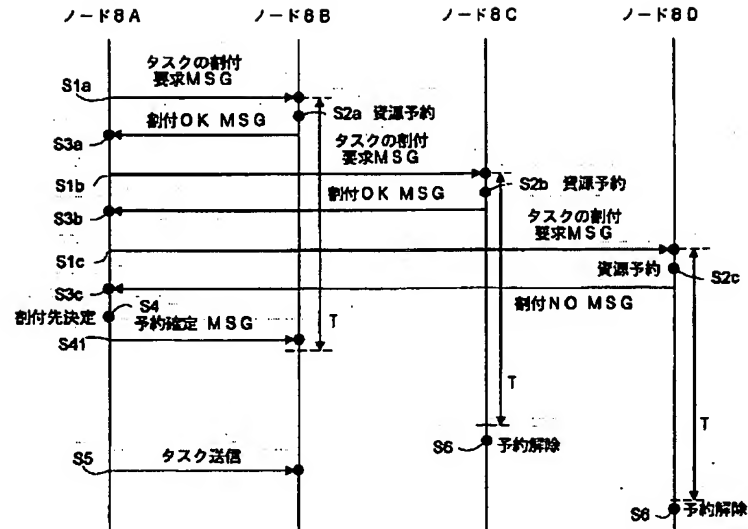
【図6】



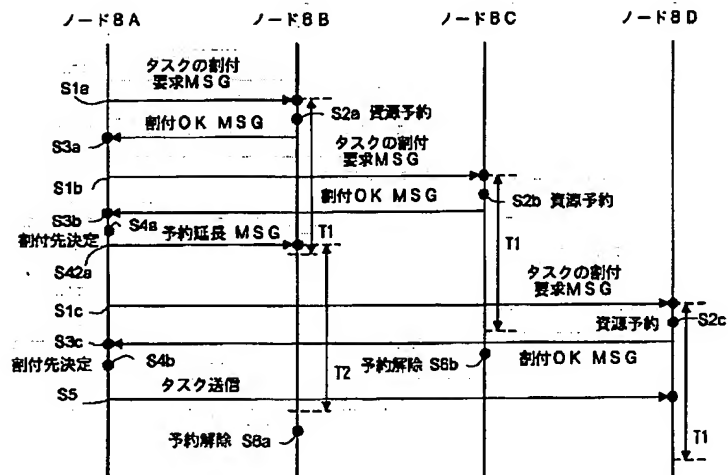
【図7】



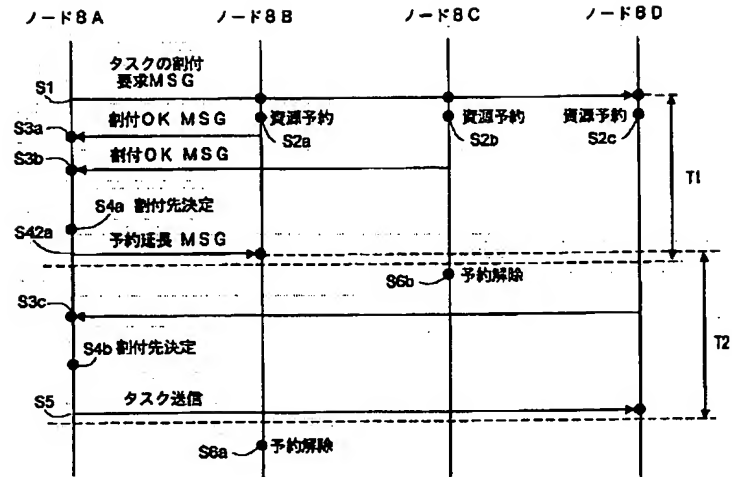
【図8】



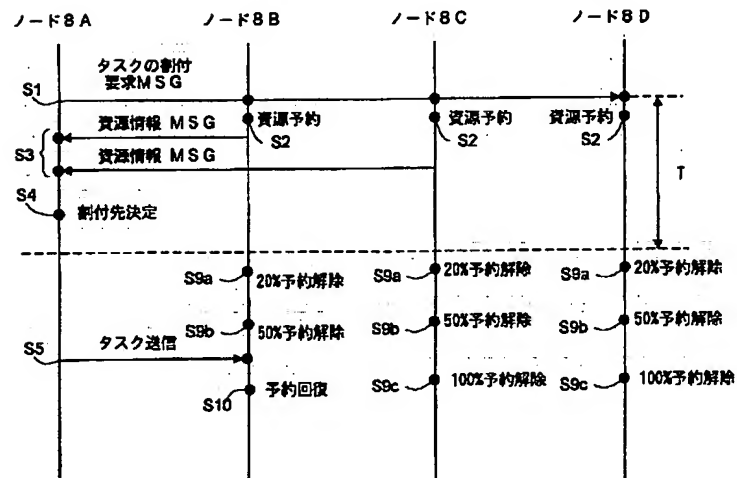
【図9】



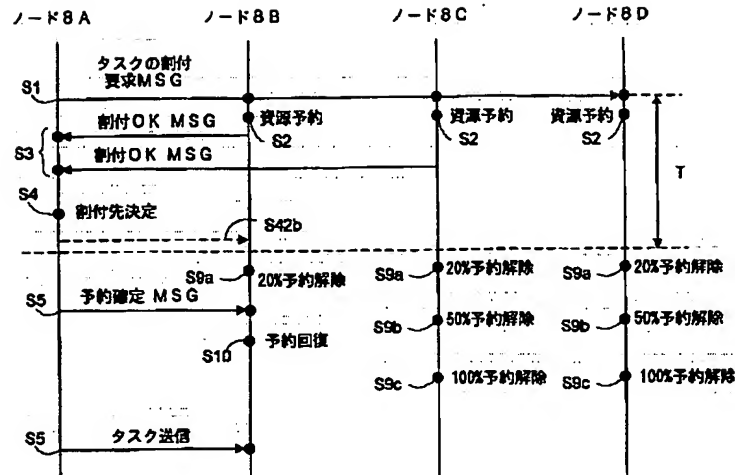
【図10】



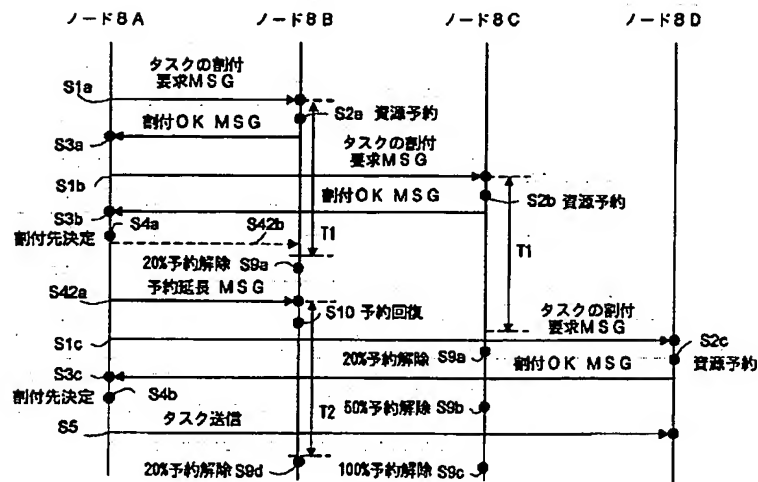
【図11】



【図12】



【図13】



【図14】

